

Highlight

Color Codes

When you are integrating your own source code into a webpage or document, syntax highlighting can provide improved readability. Highlight takes care of this task for you. **BY CHRISTIAN PERLE**

Although there are dozens of editors capable of **syntax highlighting**, only a few allow you to store formatted and highlighted document formats. *Highlight*, a purpose built tool by André Simon fills this gap, and is available at <http://www.andre-simon.de/>.

The source code archive, *highlight-1.3.tar.gz*, is also available on this month's subscription CD. As there is more to the installation than the standard rule of three *./configure; make; make install*, we have also included a **shell script** called *insthighlight.sh*.

Copy both of these files to a single directory and then type the following command to launch the script:

```
sh insthighlight.sh
```

During the course of the installation, you will be prompted to supply the *root* password as the script requires write access to the */usr/local* branch of the filesystem – to */usr/local/bin* and */usr/local/share/highlight* to be more precise.

Let there be light

To launch the program for the first time, simply use the following syntax:

```
highlight -I inputfilename?
> outputfilename.html
```

Highlight uses the file extension to recognize the input file type. Three output

OUT OF THE BOX

There are thousands of tools and utilities for Linux. "Out of the box" takes a pick of the bunch and each month suggests a little program, which we feel is either absolutely indispensable or unduly ignored.

formats are available: HTML with **Cascading Style Sheets** (CSS), Rich Text Format (RTF) and **TeX**.

The *-I* option includes the style sheet definition in the HTML output file – omitting the option will create a separate file, called *highlight.css*, in the current working directory. You can use any browser with CSS capabilities to view the output. Figure 1 shows a **POV-Ray** scene description as viewed in the Mozilla browser.

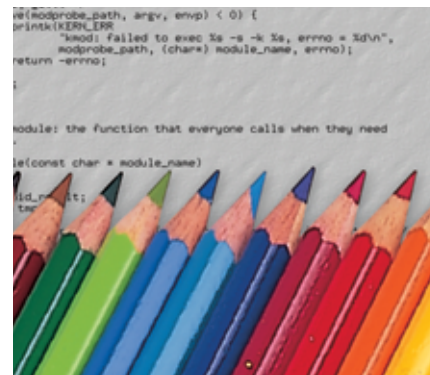
Colors and Shapes

It is quite simple to try out a different color style if the default style is not to your liking. To do so, set the *-s* flag and supply the style name. My favorites are *blue* and *darkness*. The following command provides an overview of the available styles:

```
ls /usr/local/share/?
highlight/*.style
```

You can use the following command to discover the source file types supported by Highlight:

```
ls /usr/local/share/?
highlight/*.lang
```



Attentive readers may have noticed that the language definition file for POV-ray files used in figure 1, *pov.lang*, is not listed. The good news is that the file is on the subscription CD – simply copy *pov.lang* to */usr/local/share/highlight*!

If your input file does not have an extension, you can use the *-S* option, followed by the name of the syntax definition to identify the file type, and redirect the input file from standard input using the *<* character.

As an example, Highlight will not automatically recognize your personal start script, *~/bashrc*, as a shell script as the *.sh* extension is missing. But the following syntax will do the trick:

```
highlight -I -S sh < ~/.bashrc?
> bashrc.html
```

To integrate neatly formatted code into an Office document, you will need to set the *-R* flag to specify RTF instead of HTML as the output format:

```
highlight -R inputfilename?
> outputfilename.rtf
```

You can then use the OpenOffice or StarOffice RTF import filter to import the

GLOSSARY

Syntax highlighting:

Typically the use of color-coding to emphasize specific elements in a source code or configuration file.

Shell script:

A file with shell commands that are parsed automatically. Shell scripts are typically used to automate recurring tasks.

Cascading Style Sheets: An HTML extension that abstracts concrete formatting characters from the logical structure of the document.

TeX/LaTeX:

A professional text layout system that is particularly useful for scientific documents.

POV-Ray:

A free Raytracing (3D graphics) program that runs on numerous operating systems – such as Linux. The homepage is at <http://www.povray.org/>.

Compiler Directive:

A keyword used to control the source code translation process, to omit specific code segments.

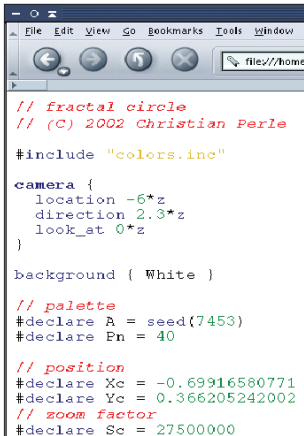


Figure 1: HTML syntax highlighting in Mozilla

output file into an existing document. The example in figure 2 shows the installation script embedded in a StarOffice document.

Users of the *TeX* layout system will be pleased to hear that Highlight can produce output in *TeX* format. The syntax is as follows

```
highlight -T inputfilename?
> outputfilename.tex
```

You can create a *PostScript* document from the *.tex* file, and then display the file with *gv* (Figure 3) or print it using *lpr*, by passing the file through the *TeX* interpreter and the *dvips* conversion tool:

```
tex outputfile.tex
dvips outputfile.dvi
```

The *-L* option is used to produce *LaTeX* output, although there was a slight glitch: a curly bracket was missing in the second to last line of the file we created.

Extending the Language Base

If Highlight is unfamiliar with the syntax of your favorite programming language, you have the option of defining a new language definition. You do not need to recompile or rewrite the program to do so; instead simply add the keywords to a language file, with the *.lang* extension and store the file in the */usr/local/share/.highlight* directory. The file format is simple:

- Lines starting with a hash sign # are comments and will be ignored.
- The **KEYWORDS =** section contains all

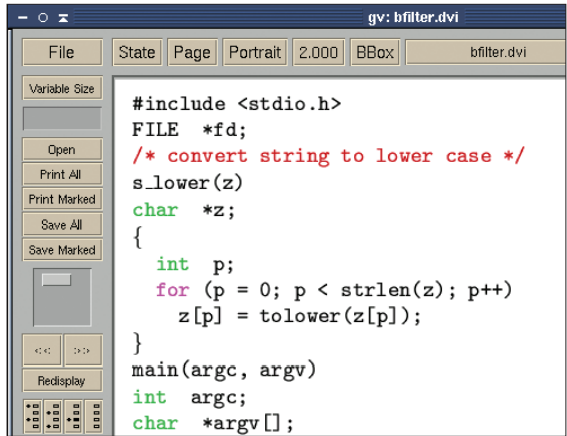


Figure 3: Producing PostScript documents means a small detour via the TeX format

the keywords supported by the language, separated by space characters. The list must be typed in a single line.

- **TYPESMODS =** is followed by a list of types supported by the language. As is the case in the keyword section, newline characters are not permitted in the list.
- **STRINGDELIMITERS =** is followed by a list of characters used to introduce or terminate strings.
- **SINGLELINECOMMENT =** is followed by the character (or string) used to designate a single-line comment.
- **MULTILINECOMMENT =** designates the character (or string) used for multiple line comments.

- **ISCASESENSITIVE =** specifies whether Highlight should (true) or should not (false) distinguish upper and lower case for keywords.
- **DIRECTIVE =** specifies the character used by the language to designate compiler directives.
- **ESCCHAR =** specifies the character used to escape non-standard characters in the language.

If the language in question does not support one of the categories in this list, you can simply omit the category. The *bash* shell does not support multiple line comments, for example, so the *sh.lang* does not include a *MULTILINECOMMENT =* entry.

Highlight's author would be pleased to receive new language definitions. So, if you are feeling bored, why not create a *.lang* file an undefined language?

THE AUTHOR

Christian Perle currently works as a developer at secunet Security Networks AG.

Christian discovered Linux in 1996, after playing around with the Sinclair ZX 81, Atari ST and finally IBM PC. When not hacking Linux stuff he can often be found playing guitar and "Magic: The Gathering".

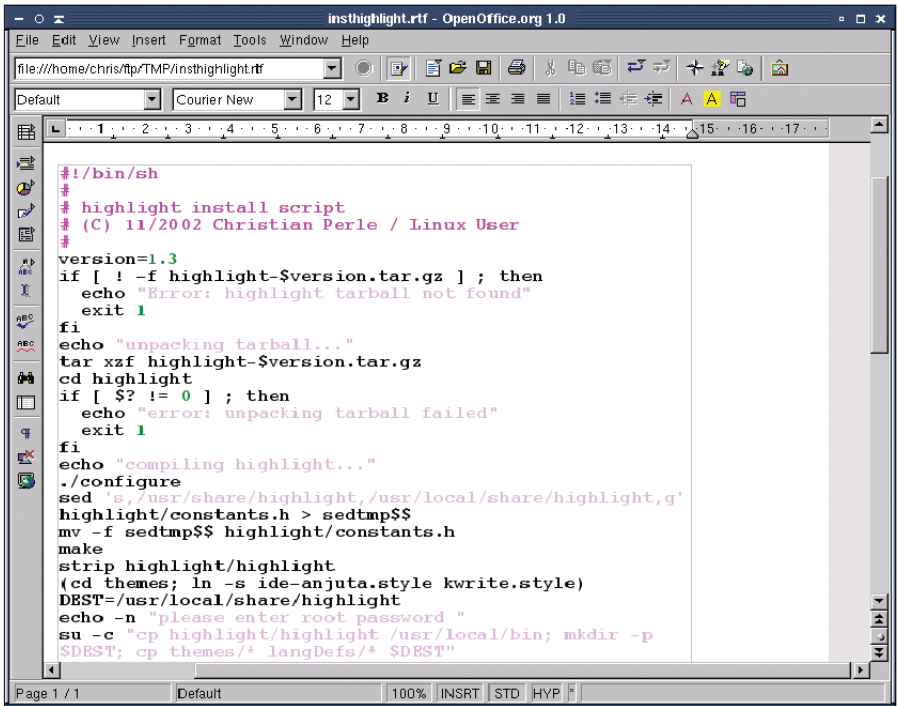


Figure 2: Neatly formatted and imported as RTF